VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2016 Bc. Matěj Vaňátko

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

WEBOVÁ APLIKACE PRO SPRÁVU SÍŤOVÝCH PRVKŮ MIKROTIK
WEB APPLICATION FOR MIKROTIK NETWORK NODES MANAGEMENT

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — **Bc. Matěj Vaňátko**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. Ondřej Krajsa, Ph.D.**
SUPERVISOR

BRNO 2016

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Diplomová práce

magisterský navazující studijní obor **Telekomunikační a informační technika**
Ústav telekomunikací

Student: Bc. Matěj Vaňátko ***ID:*** 143670
Ročník: 2 ***Akademický rok:*** 2015/16

NÁZEV TÉMATU:

Webová aplikace pro správu síťových prvků Mikrotik

POKYNY PRO VYPRACOVÁNÍ:

Navrhněte a realizujte interaktivní webovou aplikaci pro správu rozlehlé sítě založené na aktivních prvcích Mikrotik. Aplikace bude využívat Mikrotik API-SSL. Aplikaci zabezpečte pomocí uživatelských účtů a protokolu https.

DOPORUČENÁ LITERATURA:

[1] BURGESS, Dennis. Learn RouterOS. Lexington]: Dennis Burgess, 2009, 391 s. : il. ISBN 978-0-557-09271-0.

[2] LOCKHART, Josh. Modern PHP: New Features and Good Practices. Sebastopol: O'Reilly Media, 2015. ISBN 9781491905012.

Termín zadání: 1.2.2016 ***Termín odevzdání:*** 25.5.2016

Vedoucí práce: Ing. Ondřej Krajsa, Ph.D.
Konzultant diplomové práce:

doc. Ing. Jiří Mišurec, CSc., *předseda oborové rady*

ABSTRAKT

Tato práce popisuje řešení komplexní webové aplikace pro správu rozlehlých LAN a WAN sítí, které jsou založeny na aktivních prvcích společnosti Mikrotik. V práci je provedena studie proveditelnosti, která říká, jaké moduly a jakou funkčnost by měl systém obsahovat. Dále je objasněna struktura databáze a jsou nastíněny technické postupy řešení celého zadání s důrazem na univerzálnost a maximální jednoduchost.

KLÍČOVÁ SLOVA

Mikrotik, informační systém, webová aplikace, routerboard, správa, API, API-SSL, rozlehlá síť

ABSTRACT

The thesis describes a comprehensive solution of a web application for administration of extensive LAN and WAN networks, which are based on nodes by MikroTik . There is a feasibility study, which says, what modules and what functionality should be included. Also a database structure is clarified and technical procedures of solution of the whole assignment are outlined with an emphasis on universality and maximal simplicity.)

KEYWORDS

Mikrotik, information systems, web applications, routerboard, management, API, API-SSL, wide area network

VAŇÁTKO, Matěj *Webová aplikace pro správu sítě s prvky Mikrotik*: diplomová práce. Místo: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, Rok. 44 s. Vedoucí práce byl Ing. Ondřej Krajsa, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Webová aplikace pro správu sítě s prvky Mikrotik“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Místo ....

(podpis autora)

PODĚKOVÁNÍ

Rád bych poděkoval především celé mé rodině i všem mým spolupracovníkům za to, že mi umožnili na práci pracovat, za trpělivost, kterou se mnou při řešení této práce měli a také Ing. Ondřeji Krajsovi, Ph.D. za vedení práce a rychlé zodpovězení všech dotazů, které jsem při řešení měl.

Místo ....

(podpis autora)

Faculty of Electrical Engineering
and Communication
Brno University of Technology
Purkynova 118, CZ-61200 Brno
Czech Republic

http://www.six.feec.vutbr.cz

PODĚKOVÁNÍ

Výzkum popsaný v této diplomové práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno . . . . . .

. . . . . .

podpis autora

EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI

OP Výzkum a vývoj
pro inovace

OBSAH

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

ÚVOD

Mnoho velkých, ale také malých telekomunikačních operátorů, z drtivé části poskytovatelů internetového připojení, řešilo a řeší potřebu, jak správně spravovat a rozvíjet svoji telekomunikační infrastrukturu a její nastavení.

Na samém počátku tito poskytovatelé a provozovatelé sítí nikterak neřešili potřebu centrální konfigurace a ověřování vlastních zaměstnanců.

Tato práce se proto věnuje návrhu webové aplikace pro malé a střední ISP, kteří často své sítě staví pomocí aktivních prvků lotyšské společnosti MikroTik. Tyto prvky jsou přímo určené pro malé a střední operátory, ale také jako malá kancelářská či domácí zařízení. Ve světě jsou tato zařízení oblíbená pro svoji jednoduchost, nízké pořizovací náklady a pro téměř neomezené možnosti konfigurace, kterou dříve podporovaly pouze velmi drahá zařízení světových výrobců.

Aby bylo možné takovou síť operátora efektivně a hlavně rychle monitorovat, konfigurovat a spravovat, je nutné mít jeden centrální nástroj, který zajistí správnou konfiguraci každého zařízení a poskytne detailní přehled o tom, která zařízení v síti jsou, jaké konfigurace mají a umožní správně delegovat uživatelské role jednotlivých zaměstnanců poskytovatele v systému.

V této práci je rozebrán a popsán webový informační systém, který byl navržen speciálně pro sítě, které jsou postaveny z prvků MikroTik, a který operátorům sítí umožní efektivní konfiguraci svých zařízení z jednoho místa. Celý systém je postaven na open-source technologiích, jako je PHP, MySQL, JavaScript, Apache apod. Pro komunikaci s jednotlivými zařízeními pak aplikace využívá proprietární protokol MikroTik API, který je šifrován pomocí SSL.

Všechny tyto technologie jsou společně s dalšími prostředky, které byly pro řešení práce použity, detailně popsány a rozebrány v první kapitole, která je věnována teoretickým poznatkům.

Druhá kapitola se již plně zabývá vlastní programovou implementací celého systému. Jedná se tedy o hlavní kapitolu celé práce, v níž je detailně probráno všechno, čeho bylo v rámci řešení práce dosaženo. V této části tak je rozebrána samotná komunikace zařízení, jsou představeny všechny funkcionality a podobně.

Dále je zde popsán obecný návrh architektury a základní funkčnosti celého systému včetně módů, které systém podporuje. Dále je zde rozebráno nakládání s uživatelskými právy a rolemi, je osvětleno schéma databáze, představeno grafické rozhraní GUI celého systému a popsáno celkové ovládaní systému a možnosti, které system poskytuje. Také je zde nastíněna interní funkcionalita všech částí včetně zabezpečení systému a popis uvedení systému do chodu.

Třetí kapitola se věnuje detailněji ovládaní celého systému. V této části práce jsou nastíněny uživatelské postupy a práce s vlastním systémem a jeho ovládaní. Tato

kapitola tak poskytuje návod, jak se do systému přihlásit, jak přidávat, odebírat či editovat zařízení, jakým způsobem definovat jejich komunikaci a tuto komunikaci též ověřit, jakým způsobem je možné zařízení v rámci systému monitorovat a jak je možné provádět správu uživatelů a jejich uživatelských oprávnění.

V této části je také detailněji popsáno celé systémové menu a většina tlačítek včetně toho, že je detailně vysvětleno, k čemu a jak fungují.

Čtvrtá kapitola řeší primárně testování aplikace a problémy, které se při vývoji vyskytly.

Celý projekt je pak shrnut v závěru celé práce, kde jsou popsány všechny dosažené poznatky, implementované funkce a zároveň jsou zde shrnuty přednosti i nedostatky celého řešení.

1 ROZBOR A POPIS POUŽITÝCH PROSTŘEDKŮ

1.1 MikroTik

Společnost MikroTik je předním světovým dodavatelem síťových systémů a aktivních prvků převážně pro telekomunikační operátory a poskytovatele internetového připojení. Firma byla založena v roce 1995 v Lotyšsku jako výrobce levných, ale přitom spolehlivých a funkčně obsáhlých síťových zařízení pod obchodní značkou MikroTik Routerboard.

1.1.1 Routerboard

V dnešní době obsahuje nabídka společnosti výrobky určené nejen pro ISP, ale také pro malé a střední firmy, domácnosti a veřejné instituce. Její nabídka se dá rozdělit do několika oblastí, ve kterých společnost působí.

Nejdominantnějším polem působnosti je bezpochyby výroba a distribuce malých, úsporných, ale přitom velmi výkonných síťových směrovačů, které jsou dostupné pod obchodní značkou Routerboard [5]. Společnost se však nespecializuje pouze na směrovače, ale dodává na trh také síťové přepínače, prvky pro bezdrátové sítě, vysílací antény a další příslušenství.

Prvky se od sebe liší výkonem, počtem a druhy vstupně / výstupních rozhraní a portů, tvarem, vhodností umístění, modulárními vlastnostmi apod. Samozřejmě, výrobky se od sebe liší i svým určením. Některé jsou určeny do produkčního prostředí velké počítačové sítě, jiné jsou vhodné pro domácnosti a malé firmy, jiné zase pro poskytovatele bezdrátového připojení. Drtivá většina výrobků má však společný operační systém, který dodává produktům firmy MikroTik onu oblíbenost mezi technickou veřejností. Jedná se o proprietární operační systém RouterOS.

1.1.2 RouterOS

RouterOS je primárním a proprietárním operačním systémem, který běží na drtivé většině běžných zařízení firmy MikroTik . Pro přepínače je na rozdíl od směrovačů a bezdrátových prvků vyhrazen operační systém SwOS, který však není předmětem této práce a ani zařízení se systémem SwOS nejsou prozatím podporována.

RouterOS je plně modulární systém, který funguje na základě balíčků. Ty do něj přináší podporu dalších a dalších funkcí. Nepotřebné balíky tedy mohou být v případě potřeby vypnuty a je možné tak šetřit systémové prostředky zařízení.

Ovládání systému je možné prostřednictvím několika ovládacích kanálů. Jsou jimi terminál, dostupný přes SSH, Telnet či konzoli, webové rozhraní, proprietární

API rozhraní, či proprietární desktopový systém WinBox. Tato práce se zabývá převážně využitím protokolu rozhraní API v jeho šifrované podobě API-SSL. Pro řešení některých částí práce je použit také terminál a přístup k němu přes protokol SSH, konkrétně v implementaci SFTP pro přenos souborů.

Systém jako takový je určen pro všechny HW platformy značky Routerboard. Obsahuje tedy podporu pro procesory `mipsbe, smips, tile, ppc, arm a mipsle`. Velkou specialitou systému je i možnost běhu na klasických `x86` architekturách. To umožňuje spuštění tohoto jednoduchého, ale přitom robustního systému i na klasických počítačích a serverech, které disponují daleko větším HW výkonem.

Tak jako každý proprietární systém je i RouterOS licencovaný. Licence jsou rozděleny do několika stupňů, přičemž každý stupeň umožňuje vždy funkcionalitu stupně nižšího plus další funkce či rozšíření limitů. Přehledné licenční schéma je uvedeno v následující tabulce.

Tab. 1.1: Tabulka přehledu licencí systému RouterOS

Level number	0	1	3	4	5	6
Mode	Trial	Demo	WISP CPE	WISP	WISP	Controller
Initial Config Support	-	-	-	15 days	30 days	30 days
Wireless AP	24h	-	-	yes	yes	yes
Wireless Client	24h	-	yes	yes	yes	yes
Wireless Bridge	24h	-	yes	yes	yes	yes
RIP, OSPF, BGP	24h	-	yes(*)	yes	yes	yes
EoIP tunnels	24h	1	unlimited	unlimited	unlimited	unlimited
PPPoE tunnels	24h	1	200	200	500	unlimited
PPTP tunnels	24h	1	200	200	500	unlimited
L2TP tunnels	24h	1	200	200	500	unlimited
OVPN tunnels	24h	1	200	200	unlimited	unlimited
VLAN interfaces	24h	1	unlimited	unlimited	unlimited	unlimited
HotSpot active users	24h	1	1	200	500	unlimited
RADIUS client	24h	-	yes	yes	yes	yes
Queues	24h	1	unlimited	unlimited	unlimited	unlimited
Web proxy	24h	-	yes	yes	yes	yes
User manager sessions	24h	1	10	20	50	unlimited
Number of KVM guests	-	1	unlimited	unlimited	unlimited	unlimited

1.2 MikroTik API

1.2.1 Obecný popis protokolu

MikroTik API je proprietární komunikační programové rozhraní společnosti MikroTik pro komunikaci prvků s operačním systémem RouterOS s dalšími síťovými zařízeními a aplikacemi.

Toto rozhraní vzniklo z důvodu standardizace komunikace systému RouterOS s dalšími systémy a službami. API rozhraní dovoluje nejen výpis a sběr informací o zařízení a jeho komponentách, ale také přímou úpravu konfigurace, správu samotného systému a možnost systém kdykoliv restartovat.

Stejně jako další možnosti konfigurace, je i API rozhraní plnohodnotným prostředkem ke správě routeru s běžícím RouterOS. Poskytuje totiž stejné možnosti konfigurace a správy, jako proprietární webové rozhraní či nástroj WinBox.

1.2.2 Syntaxe příkazů

Syntaxe příkazů API je navíc velmi podobná syntaxi příkazů klasické příkazové řádky. Lze tedy velmi jednoduše vytvořit nástroj, který převede příkaz příkazové řádky na API příkaz a obráceně.

Umožňuje to také velmi jednoduše vystavět nad API robustní systém, který právě využívá úrovňovost a šablonovitost všech příkazů RouterOS. Tento přístup byl zvolen jako hlavní při řešení této práce.

1.2.3 Komunikace

Komunikace přes API rozhraní je opět velmi jednoduchá a snadno implementovatelná. Probíhá vždy tak, že API klient zašle sekvenci slov, tedy příkaz, na který očekává od serveru odpověď. Každý příkaz je složen ze slov a je ukončen slovem s nulovou délkou. Jakmile router obdrží celý příkaz a všechny či žádné atributy, je příkaz dekódován, proveden a jeho výstup je zaslán jako odpověď klientovi.

Standardně probíhá komunikace s API rozhraní přes TCP port 8728. Tato komunikace je nešifrovaná a jako ověření je použito jméno a heslo, které je společné pro všechny komunikační kanály. Volitelně lze zapnout podpora pro šifrovaný API přenos pomocí SSL, který využívá port TCP 8729 a který bude také použit v rámci řešení této práce. Komunikace pomocí API-SSL bude v této práci využívat klasické ověření pomocí certifikátů. V nastavení zařízení je tedy nutné takový certifikát vygenerovat a API rozhraní přiřadit.

Tab. 1.2: Kódování délky slova API rozhraní

Délka slova	Počet Bytů	Kódování
0 <= délka <= 0x7F	1	délka, nejnižší Byte
0x80 <= délka <= 0x3FFF	2	délka | 0x8000, dva nejnižší Byty
0x4000 <= délka <= 0x1FFFFF	3	délka | 0xC00000, tři nejnižší Byty
0x200000 <= délka <= 0xFFFFFFF	4	délka | 0xE0000000
délka >= 0x10000000	5	0xF0 délka jako čtyři Byty

1.2.4 Kódování slov API

Každé slovo API rozhraní je kódováno samostatně. Slovo je vždy kódováno tak, že na začátku slova je zakódována jeho délka v Bytech, které tvoří samotný obsah slova a jsou bezprostředně připojena za tento údaj. Délka slova je tedy vždy počet Bytů, které budou odeslány a obsahují vlastní data. Kódování je provedeno vždy podle následující tabulky manuálové stránky API[3].

1.3 PHP a MySQL

1.3.1 Obecný popis PHP a jeho použití v projektu

Jazyk PHP je moderní skriptovací jazyk, vhodný zejména pro psaní dynamických webových stránek a aplikací. Současně s tím lze použít také jako jazyk pro tvorbu konzolových i desktopových aplikací. Toto použití je však minoritní, oproti využívání jazyka v prostředí internetových stránek.

PHP je dnes nejrozšířenějším jazykem pro psaní webových aplikací na světě s podílem více než 80%.

Všechny skripty PHP jsou prováděny vždy na straně serveru. Ke klientovi je přenášen až pouze jejich výsledek. Jazyk je naprosto nezávislý na použité platformě systému, na kterém běží. Jeho interpret podporuje jak operační systém Windows, tak operační systém Linux a mnoho dalších. Aplikace napsané v PHP lze tak většinou bez problémů přenášet mezi různými systémy. I z tohoto důvodu bylo PHP vybráno jako jazyk pro tvorbu tohoto projektu.

Syntaxe jazyka je velmi blízká jiným, mnohem starším jazykům, jako je například Pascal či jazyk C. Struktura jazyka je opět modulární a podporuje mnoho knihoven pro různé účely. V tomto projektu je použita hlavně podpora pro databázový systém MySQL.

Obvyklé nasazení PHP je v rámci tzv. balíčku zkzkLAMP. Tedy v kombinaci s operačním systémem Linux, webovým serverem Apache a databázovým systémem MySQL. LAMP bude použit pro běh i v případě tohoto projektu.

Hlavními důvody, proč byl pro realizaci projektu vybrán právě jazyk PHP, je hlavně jeho specializace přímo na webové stránky, nativní podpora systému MySQL, obrovské množství funkcí a modulů, multiplatformnost, velká dostupnost hostingových služeb, jednoduché nasazení v balíčku LAMP, velké množství veřejně dostupných kódů zdarma, svobodná licence využití jazyka, aktuální vývoj a vynikající dokumentace dostupná online[2].

Jednou z mála nevýhod je, že jazyk ve standardní distribuci neobsahuje žádný robustní ladící nástroj, který by pomohl s hledáním chyb v programové implementaci. Tento nedostatek se také několikrát projevil i při psaní tohoto projektu.

1.3.2 Databázový systém MySQL

Databázový systém MySQL je dnes, stejně jako jazyk PHP, jedním z nejrozšířenějších databázových nástrojů používaných při tvorbě webového obsahu. Samotné MySQL je vlastněno a vyvíjeno švédskou společností MySQL AB, která je vlastněna firmou Sun Microsystems. Obě dvě společnosti jsou však v majetku firmy Oracle Corporation.

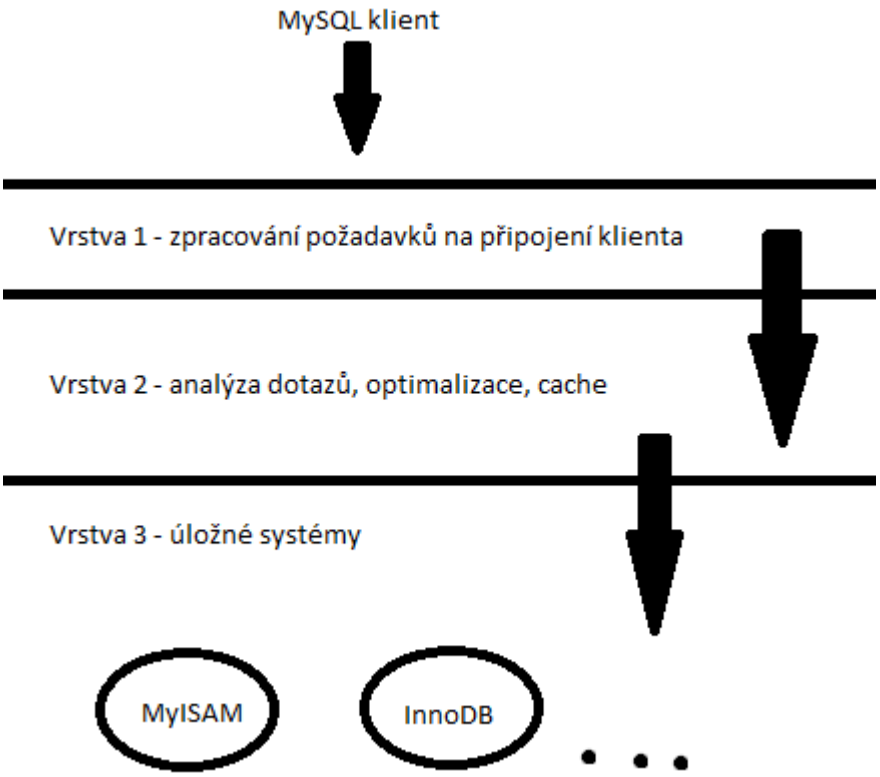
Od začátku je MySQL databáze vyvíjena především s důrazem na rychlost. Této vlastnosti bylo dosaženo nejen optimalizací celého kódu databáze, ale také především celkovým zjednodušením. Nepodporovalo pohledy či uložené procedury a má jen omezené možnosti zálohování.

Naopak, jeho velkou výhodou je opět jeho modularita a hlavně multiplatformnost. Systém je použitelný opět na většině současných operačních systémů, avšak, největší obliby se těší, stejně jako jazyk PHP, v balíčku LAMP. Další nespornou výhodou databázového systému MySQL je jeho široký záběr, který umožňuje jeho použití v nespočtech různorodých projektech. Toho je dosaženo rozdělením architektury systému do několika vrstev.

První vrstva obsahuje obslužné rutiny a funkce pro komunikaci s databázovými klienty prostřednictvím sítí a soketů. Tato vrstva je společná defacto pro všechny databázové systémy.

Většina operací systému se děje na druhé vrstvě, na které se nachází samotný kód pro rozbor příkazů, analýzu a optimalizaci. Tato vrstva obsahuje také cache dotazů, jež je využívána pro často se opakující dotazy, jejichž vyřízení by trvalo zbytečně dlouho, pokud by měla být volána třetí vrstva.

Ta obsahuje už samostatné úložné systémy. Ty se liší svými možnostmi použití a způsobem ukládání dat do souborů. Nejvýznamnějším z nich jsou úložiště MyISAM a InnoDB. Přehledné schéma architektury MySQL je zobrazeno na 1.1



Obr. 1.1: Architektura vrstev systému MySQL

2 VLASTNÍ PROGRAMOVÁ IMPLEMENTACE PROJEKTU

2.1 Obecný popis řešeného problému a cíl práce

Cílem této diplomové práce bylo vytvoření jednoduché webové aplikace, která by byla silným nástrojem a pomocníkem při správě počítačové sítě, která je tvořena převážně aktivními prvky společnosti MikroTik a kterou by ocenili nejen poskytovatelé služeb na bázi ISP, ale také střední podniky, státní instituce a IT outsourcingové společnosti a to nejen jako nástroj pro správu jejich lokálních sítí na bázi prvků MikroTik , ale také jako nástroj pro správu klientských a zákaznických zařízení daných firem.

Obsahem bylo nejen provést analýzu celé skutečnosti, ale především také vlastní programová implementace celého projektu, její odladění a otestování a také navržení a implementace kvalitního grafického rozhraní, které by bylo nejen hezké na pohled, ale také přehledné a uživatelsky intuitivní.

2.1.1 Analýza problému

Prvně bylo nutné provést analýzu zařízení firmy MikroTik , jeho architektury a funkcí a bylo nutné provést detailní analýzu způsobu komunikace skrze protokol MikroTik API. Bylo zjištěno, že konfigurace mezi jednotlivými zařízeními je velmi jednoduše přenositelná, neboť příkazová řádka má stále ten samý tvar. Při přenosu konfigurace se tedy nekompatibilní část vynechá, ale zbylá nastavení se přenesou v pořádku. Je to dáno tím, že operační systém RouterOS je modulární. Umožňuje tak běh pouze těch částí, které dané zařízení podporuje v rámci svého hardware. Nemůže se tedy stát, že zařízení, které v sobě nemá například WiFi kartu, by přijímalo příkazy, které jsou určeny pro zařízení, která WiFi modul mají. Toto už hlídá sám RouterOS.

2.1.2 Části systému

Jakmile byla dokončena analýza problému a zadání, bylo nutné udělat správnou vnitřní architekturu systému. Zde byl kladen velký důraz na jednoduchost, ale přitom s maximálním zabezpečením a rychlostí celého systému. Jako jádro byl zvolen robustní jazyk PHP. Na toto jádro systému je poté napojen databázový systém MySQL, grafické uživatelské rozhraní a všechny obslužné funkce API rozhraní. Samotný systém pro komunikaci s API rozhraním je velice obsáhlý, ovšem jeho návrh

je velmi jednoduchý a využívá elegance příkazové řádky operačního systému RouterOS. Díky tomu je podpora zařízení firmy MikroTik v systému takřka stroprocentní.

2.2 Přehled funkcí systému

2.2.1 Obecná funkcionalita

Celý projekt je od začátku kompletně koncipovaný a vyvíjený jako modulární webový informační systém, ktery plně využívá elegance struktury příkazů operačního systému zařízení RouterOS. Oproti semestrálnímu projektu však došlo k práci celkem k dramatickým změnám, co se týče podporovaných příkazů. Ve verzi semestrálního projektu podporoval systém pouze některé základní balíčky RouterOS. Byly to balíčky `/ip, /interface a /queue`.

Nicméně, celý systém je navržen tak, aby maximálně využíval eleganci stromového tvaru přikázů systému RouterOS. Veškeré možné příkazy jsou uchovávány v rámci šablony v jedné z tabulek systémové MySQL databáze. Pouhým přidaním definice totoho příkazu do databáze je docílena plná podpora příkazu v rámci celého systému, ať už se jedná o uživatelská práva, grafické uživatelské rozhraní či všechny potřebné editační a vkládací formuláře.

Z toho vyplývá, že grafika systému i jeho vlastní funkcionalita je modulární a je editovatelná jednoduchými změnami v databázi. Nicméně, systém nepodporuje pouze šablony příkazů zadané uživatelem. Systém sám si dokáže vyčíst aktuální konfiguraci zařízení a tuto konfiguraci nejenže okamžitě ukládá a zařízení tak zařazuje do systému, ale zároveň vytváří v databázi šablony příkazů, které systém ještě nezná. Tato funkce tvoří systém velmi jednoduchým a lehce přizpůsobitelným k jakémukoliv balíčku operačního systému RouterOS.

V současné době systém podporuje všechny nutné funkcionality, které systém potřebuje pro pohodlnou správu všech zařízení MikroTik . Aplikace umí přidat, editovat či odebrat jakýkoliv router, ke kterému udržuje přihlašovací údaje pro spojení a konfiguraci, eviduje GPS souřadnice zařízení, IP adresu na které se nachází a vede k routeru statistické i konfigurační údaje. Z každeho zařízení systém automaticky vyčítá skrze SNMP jeho provozní údaje, jako je obsazenost paměti RAM, vytížení procesoru či odezva na zařízení. Nejdůležitějším údajem o každém zařízení je však udržovaná konfigurace.

2.2.2 Podporovaná zařízení

Mezi podporovaná zařízení se dají zařadit všechna zařízení, která společnost MikroTik vyrábí a prodává a jejichž základ tvoří operační systém RouterOS. Systém totiž

využívá elegantní vlastnosti celého prostředí a to je univerzálnost příkazové řádky, resp. příkazů API rozhraní a balíčkovací systém. Z toho plyne několik jasných výhod. Konfigurace je nejen lehce přenositelná, ale je také přenositelná a použitelná na všechna zařízení s RouterOS.

V současné době tak systém podporuje Routerboardy řad RB3xx, RB600, RB5xx, RB1xx, RB800, RB1000, RB1100, RB4xx, RB7xx, RB9xx, cAP, mAP, hEX, RB2011, hAP lite a počítače s architekturou x86. Všechna tato zařízení mohou být bez problémů do systému zapojena a ze systému spravována. Jak je vidět, jsou mezi nimi zařízení určená pro malé sítě v domácnostech či kancelářích – typicky řada RB9xx, ale také zařízení pro velké sítě a páteřní linky, jako je například série RB1100, která vyniká svým výkonem.

Podpora zařízení je dána dvěma faktory. Tím prvním je znalost všech jeho možných konfiguračních parametrů. Jinými slovy, znalost všech šablon příkazů, které dané zařízení podporuje. Druhým faktorem je existence typu zařízení v systému. Systém umožňuje definici jednotlivých typů zařízení podle toho, ktere příkazy daný typ zařízení podporuje.

2.2.3 Správa podporovaných příkazů

Velmi elegantně je v práci i v systému řešena vlastní funkčnost a podpora všech různých příkazů systému RouterOS. Nikde v PHP kódu aplikace totiž není definováno, jak má příkaz pro API rozhraní RouterOS vypadat, jaké akce dané nastavení umožňuje, zda je možné ho pouze editovat či je možné entity přidávat i odebírat a jaké hodnoty vlastně má uživatel zadávat, aby byl příkaz validní a verifikovatelný. Naopak, je plně využito elegance stromové struktury příkazů a veškeré kontroly a práce s příkazy jsou ověřovany a dělány skrze šablony příkazů v databázovém systému. Zde byla využita jistá úroveň příkazů a rozdělení jejich skladby do několika částí. Typický příkaz se totiž skládá takto:

- Úroveň zanoření
- Požadovaná akce
- Parametry a jejich hodnoty

Tento přístup je stejný pro všechny balíky RouterOS. To umožnilo vznik plně dynamického systému, který není nutné upravovat po stránce PHP kódu, ale stačí správným způsobem vložit správné údaje do MySQL databáze. Do databáze se vloží pouze příkaz v maximální tvaru, tedy včetně všech volitelných nastavení a parametrů. Pokud je požadavek na přidání nového záznamu, je dynamicky vygenerován správný formulář se všemi políčky, které příkaz obsahuje a které je možno vyplnit. To, o jaký příkaz se jedná, říká právě úroveň zanoření. Pokud se bude jednat například o `/ip address`, je jasné, že se vygenerují políčka pro přidání IP adresy na

rozhraní. Stejně tak tomu je i v případě úpravy IP adresy. To se od přidání liší pouze tím, že ve výsledku není použit příkaz `/ip address add` – tedy akce ADD – ale je použit příkaz `/ip address set`.

Každý příkaz, jak je výše uvedeno, se skládá nejen z úrovně zanoření a parametrů, ale také z požadované akce, kterou si přeje administrátor podniknout. Tyto akce jsou děleny následovně:

- ADD - akce, která přidává nějakou konfiguraci do systému. Například výše uvedené přidaní IP adresy na nějaké komunikační rozhraní
- SET - akce, která nějakým způsobem mění některý z parametrů již existujícího příkazu případně vestavěné HW komponenty. Typickým příkladem zadaného příkazu je změna IP adresy s daným identifikátorem. Naopak, krásným příkladem nastavení HW komponenty je úprava taktování procesoru, kdy je pomocí akce SET nastavena požadovaná frekvence systémového procesoru.
- REMOVE - akce, jenž maže některý uživatelsky zadaný příkaz. Nedá se tak použít pro vestavěné komponenty, jako je například zmiňovaná frekvence procesoru.

Tyto příkazy slouží primárně pro změnu konfigurace v tom smyslu, že je přidána, mazána či měněna. Nicméně, jsou měněny pouze parametry daného příkazu. Není možné příkaz dočasně zneaktivnit nebo opětovně povolit. K tomu slouží následující dvojice akcí:

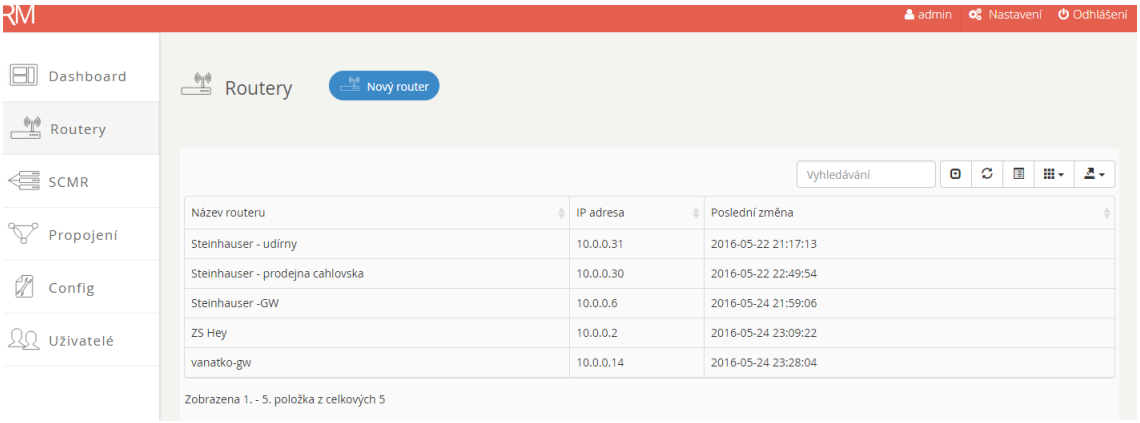
- DISABLE - akce, která umožňuje zneaktivnit již zadaný příkaz. Rozdíl oproti akci REMOVE je však ten, že akce DISABLE příkaz neruší trvale, ale pouze ho zneaktivní. Příkaz tak v systému sice existuje, ale systém na něj nijak nereaguje a při běhu systému se nijak neuplatňuje.
- ENABLE - opak akce DISABLE, která neaktivní příkaz opět povolí. Po povolení příkazu na něj systém opět začne brát ohled a začne příslušným způsobem ovlivňovat funkčnost zařízení.

Akce `SET, REMOVE, ENABLE a DISABLE` však vyžadují pro svoji správnou funkčnost znalost toho, co se má editovat. K tomu slouží v RouterOS systém vnitřních ID. Každý záznam má vždy ID 0 a vyšší. Pokud se však v řadě čísel jedno ID smaže, všechny ostatní vyšší jsou o 1 dekrementovány. Tohoto je docíleno pomocí dynamického zjišťování čísla ID při jakémkoliv požadavku na jakoukoliv změnu zadaného příkazu. Systém se dotáže daného zařízení na danou konfiguraci a z ní pozná, jaké je aktuální ID daného příkazu v ramci systému. Nemělo by tedy vadit upravovat konfiguraci zařízení jak skrze tento systém, tak napřímo v RouterOS pomocí například aplikace WinBox.

2.2.4 Přidání zařízení do systému

První funkcí, kterou bylo nutné realizovat, bylo přidání nového zařízení do systému a jeho inicializace. Samotný proces přidání zařízení je velmi jednoduchý. Složitá je pouze samotná inicializace. Zařízení je do systému přidáno pomocí průvodce, ve kterém je vyplněn jeho název, IP adresa, uživatelské jméno lokálního uživatelského účtu v zařízení a jeho heslo. Pomocí toho účtu pak systém se zařízením komunikuje jak pomocí API, tak pomocí protokolu SFTP. Dále je možné při zadaní zařízení přidat jeho GPS souřadnice, které pak ukazují zařízení na vygenerované mapě a také je možno přidat krátkou poznámku, která zařízení více specifikuje a popisuje. Speciální možností je přidaní tzv. tagu. Tag má stejnou funkcionalitu jako skupiny a bude dále rozebrán v další části práce.

Jakmile je zařízení přidáno, je možnost, aby z něj byla vyexportována jeho současná konfigurace, která je poté přidána do systému. Díky tomu je vytvořen obraz konfigurací 1:1, kdy konfigurace v MySQL databázi přesně odpovídá reálné konfiguraci na zařízení. Konfigurace je exportována tak, že prostřednictvím API rozhraní je zařízení vyzváno k exportu současné konfigurace do souboru. Export je prováděn i s parametrem verbose, který jako jediný prozradí úplnou konfiguraci zařízení. To je nutné k detekování výchozích nastavení, hlavně k detekování HW vstupně-výstupních portů. Samotný vygenerovaný soubor je pak prostřednictvím protokolu SFTP stažen a ihned zpracován přímo pomocí PHP. Exportem konfigurace je přidání zařízení dokončeno a mohou být na něj dále aplikovány další konfigurační změny, které jsou tentokrát inicializovány ze strany webové aplikace. Bohužel, z dostupných materiálů nebylo možné zjistit, zda přenos souborů podporuje samotné API rozhraní. Proto pro přenos souboru byl zvolen protokol SSH a jeho varianta SFTP, která je zde jako nejjednodušší řešení.



Obr. 2.1: Přehled přidaných zařízení do systému.

Aby bylo možné zařízení fyzicky do systému přidat, je nutné na zařízení udělat některé kroky ještě mimo tento informační systém a to například prostřednictvím aplikace WinBox či přes webové rozhraní zařízení. První nutnou konfigurací je nastavení uživatele, skrze kterého se bude systém do zařízení přihlašovat. Toho je docíleno skrze příkaz `/user add name=JMENO password=HESLO group=full`, kde JMENO je nahrazeno za požadované uživatelské jméno a HESLO je nahrazeno za heslo, kterým se má systém vůči zařízení autorizovat. Tento příkaz tak umožní vzdálené přihlášení k zařízení. Nicméně, aby fungovalo správně šifrovaní rozhraní API-SSL pro přístup, je nutné vygenerovat SSL certifikát zařízení a importovat ho pro službu API-SSL. Toho je možné docílit prostřednictvím těchto příkazů: `/certificate add common-name=JMENO name=JMENO`, kde JMENO je požadované jméno certifikátu `/certificate sign name=JMENO numbers=0`, kde JMENO je opět jméno certifikátu. Parametr numbers=0 očekává, že vygenerovaný certifikát byl prvním certifikátem, který v zařízení existuje. Zbývá tedy pouze vygenerovaný certifikát přiřadit službě API-SSL. To je docíleno pomocí následujícího příkazu: `/ip service set certificate=JMENO api-ssl`, kde JMENO je opětovně jméno vygenerovaného certifikátu.

Po těchto úpravách by mělo být zařízení plně připraveno k tomu, aby bylo se systémem funkční a bylo možné konfigurace editovat. Správnost konfigurace signalizuje zelený smajlík OK, který je poté u zařízení uveden, pokud je v systému aktivní.

2.2.5 Import konfigurace zařízení do systému

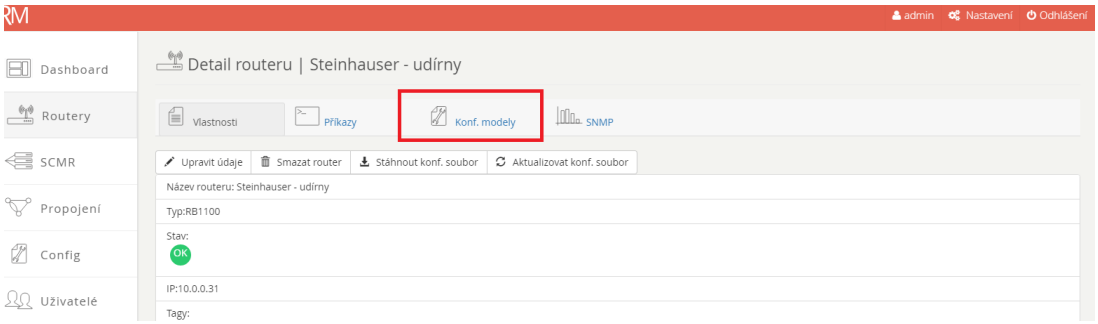
Jak již bylo řečeno výše, systém umožňuje import aktuální konfigurace. Import není automaticky prováděn ihned po přidání zařízení, ale na vyžádaní uživatele systému. Import samotný je plně automatizovaný. Stačí, aby uživatel systému stiskl u routeru tlačítko pro aktualizaci konf. souboru a systém provede všechno ostatní. Stahování je signalizováno točícím se kolečkem. Jakmile je stažení provedeno, je o tom uživatel ihned obeznámen.

2.2.6 Stažení konfiguračního souboru

Vedle tlačítka pro aktualizaci konfigurace je možné také stažení samotného konfiguračního souboru. To se hodí pro místa, která není možné do systému přidat a přesto je zde požadavek na centrální a jednotnou konfiguraci daných zařízení. Pomocí tohoto tlačítka je stažen textový soubor konfigurace, který lze jednoduše spustit jako skript na zařízení, který samotnou konfiguraci provede.

2.2.7 Režim šablonování konfigurace

Prvním možným konfiguračním režimem je režim, který zařízení konfiguruje pomocí konfiguračních vzorů, takzvaných šablon. Tyto šablony umožňují administrátorům velmi elegantně a rychle konfiguraci většího množství zařízení tím, že na ně pouze aplikují jednu připravenou šablonu a nemusí tak konfigurovat každé zařízení zvlášť. Tento režim je ten, který se používá pro velké konfigurační změny a umožňuje kompletně změnit konfiguraci daného zařízení. Je použit tak, že zařízení je nejdříve zapojeno do systému výše zmíněnou procedurou a poté je jeho konfigurace kompletně přepsána konfiguračním vzorem. Přepsání je vyvoláno tak, že vygenerovaný konfigurační soubor je opět pomocí protokolu SFTP přenesen na požadované zařízení, které je poté pomocí API příkazu resetováno do továrního nastavení. Tento příkaz je spuštěn s parametry `no-defaults=yes skip-backup=yes run-after-reset=[nazev vygenerovaneho souboru konfigurace]`. Tím je docíleno to, že zařízení má pouze tu konfiguraci, kterou správce chtěl. Typickým příkladem použití tohoto přístupu je výměna jednoho prvku v síti za druhý. Aktivaci konfiguračního vzoru je vidět na následujícím obrázku.



Obr. 2.2: Aktivace a nasazení konfiguračního vzoru či aktualizace zařízení na základě vzoru.

2.2.8 Režim hromadných příkazů

Oproti semestrálnímu projektu, kde byl režim hromadných příkazů řešen pomocí konfiguračních vzorů, došlo v této verzi práce k velkému přepracování. Přístup skrze konfigurační vzor byl kompletně zrušen a byla vytvořena možnost hromadného přidání příkazů na více zařízení. Pro tento přístup je v hlavním menu aplikace ikona SCMR - single command to multiple routers. Tento přístup je daleko jednodušší, protože hromadných změn není tolik, aby bylo nutné pro každou změnu generovat nejdříve konfigurační vzor a pak ho teprve aplikovat.

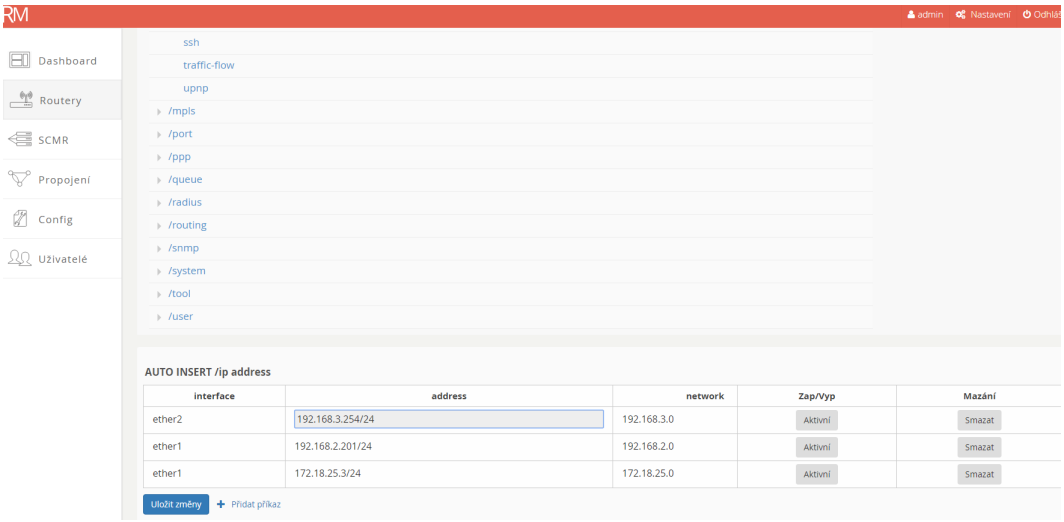
Tento přístup umožňuje aplikaci nového příkazu či přepsání původního příkazu okamžitě bez zbytečného zdržování. Stačí pouze v systému vybrat zařízení, která mají být změnou ovlivněny a jednotlivé příkazy, které mají být zadány. Jakmile jsou údaje vyplněny, jsou příkazy paralelně provedeny na všech zařízeních.

2.2.9 Režim úprav v reálném čase

Další funkcí systému jsou úpravy konfigurace v reálném čase. Specifikem této funkcionality je, že změny jsou v zařízení prováděny téměř okamžitě, jakmile jsou provedeny ve webovém rozhraní systému. Zdržení je pouze v rámci komunikace skze API-SSL rozhraní. Upravená konfigurace je zapsána nejen do zařízení, ale také do databáze systému, aby byl zachován přesný obraz 1:1 konfigurace zařízení a konfigurace uložené v databázi. Zapsaná konfigurace je na zařízení nastavena ihned po provedení změny. Přenos konfiguračních souborů již není vůbec realizován prostřednictvím protokolu SFTP, ale pouze pomocí rozhraní API. Všechny změny, které se provádějí v reálném čase probíhají pouze skrze API. Žádné soubory ani přenos skrze SFTP tak realizován není.

Jakmile uživatel provede jakoukoliv změnu, je na základě databáze příkazů vygenerován správný příkaz, který danou změnu na zařízení provede a přes API rozhraní je vykonán. Jestliže je příkaz proveden v pořádku, je konfigurace uložena také do databáze systému. Pokud však při provádění příkazu dojde k jakékoliv chybě, je o tom uživatel informován chybovou hláškou, zápis do databáze proveden není a příkaz není vykonán. Zároveň s tím je tato chyba zaznamenána i do databáze.

Režim konfigurace v reálném čase je dostupný pouze při konfiguraci konkrétního zařízení. Uživatel má k dispozici webové rozhraní, které je velmi podobné prostředí programu WinBox. Z tohoto přístupu plyne stejná nevýhoda, jako v případě použití nástroje WinBox. Neodborným zásahem do konfigurace zařízení může dojít ke ztrátě spojení a zařízení již nemusí být dále konfigurovatelné. Zařízení v tomto stavu pak vyžaduje fyzický zásah administrátora. Toto chování je zapříčíněno bohužel díky tomu, že aplikace prozatím nepodporuje funkci safe-mode, kterou bohužel dle dostupných materiálů nepodporuje ani samotné API-SSL rozhraní.



Obr. 2.3: Úprava konfigurace v reálném čase.

2.2.10 SNMP grafy

Další funkcionalitou systému je statistická a přehledová funkce díky generování a zobrazování SNMP dat. Pro každé zadané zařízení, které má povoleno i SNMP, generuje systém několik grafů provozu. V současné době systém generuje každou minutu graf typu XY a to pro hodnoty vytížení CPU, množství využité RAM a odezvu na zařízení. Nicméně, opět jednoduchým zásahem do databáze, konkretně do tabulky snmpid, je možno jakýkoliv graf opět přidat. Stačí přidat, co se má měřit a jaké snmp OID veličina má. Systém ji pak okamžitě začne měřit.

OID je podporováno opět v několika formách. Buď je zadáno pouze samotné OID, což vede k tomu, že bude vygenerován klasický graf jedné veličiny. Pokud jsou zadány 2 OID, které jsou odděleny středníkem, jsou hodnoty druhého OID v pořadí odečteny od hodnoty prvního OID a výsledek je zobrazen v grafu. Třetí možnost je relativní vyjádření tohoto výsledku, pro což stačí přidat třetí OID, který nebude v klasickém tvaru OID, ale bude mít tvar pouze znaku procenta. Potom je rozdíl prvních dvou OID převeden na relativní a je vyjádřen v procentech. Speciálním OID je pouze slovo "echo". Tím pádem je generován graf odezvy, kdy je vyslán ping a jeho hodnota je zaznamenána do databáze. V současné době je grafování povoleno pro všechna zařízení a je to tak zabudovaná a neměnná vlastnost.

V ramci PHP byl použit balíček php5-snmp. Pro správné fungování je nutné spouštět skript, který zařízení kontroluje a vyčítá z nich údaje, každou minutu pomocí plánovače úloh či nějakého cronu. Fyzicky je nutné každou minutu spouštět skript `./public/snmpget.php`, který se kompletně o vyčítání postará.

2.2.11 Graf propojení

Další novinkou oproti semestrálnímu projektu je graf propojení zařízení, které jsou obsaženy v systému. Jedná se o plně dynamicky generovaný graf, který ukazuje, jak jsou zařízení, která jsou v systému obsažena, navzájem propojena. Na grafu jsou vidět nejen samotná zařízení, která jsou v systému, ale jsou zde zobrazeny i orientované šipky, které ukazují, jak je propojení realizováno. Jak již bylo řečeno, graf je generován plně automaticky. Toho je docíleno vyčítáním diagnostické funkce RouterOS, která zjišťuje sousedy zařízení. Jedná se o funkcionalitu `/ip neighbor`, díky níž systém vyčte sousední zařízení. Pokud najde zařízení, která na sebe vidí, vygeneruje je v grafu a orientovanou šipkou označí, které zařízení vidí které. Je tak možné vidět i to, že jedno zařízení vidí druhé, ale druhé nevidí první, protože chybí šipka ve druhém směru. Tento graf je tak i velmi dobrým diagnostickým nástrojem, kterým lze spolehlivě zjistit některé typy závad při komunikaci v rámci sítě.

2.2.12 Nastavení uživatelských práv

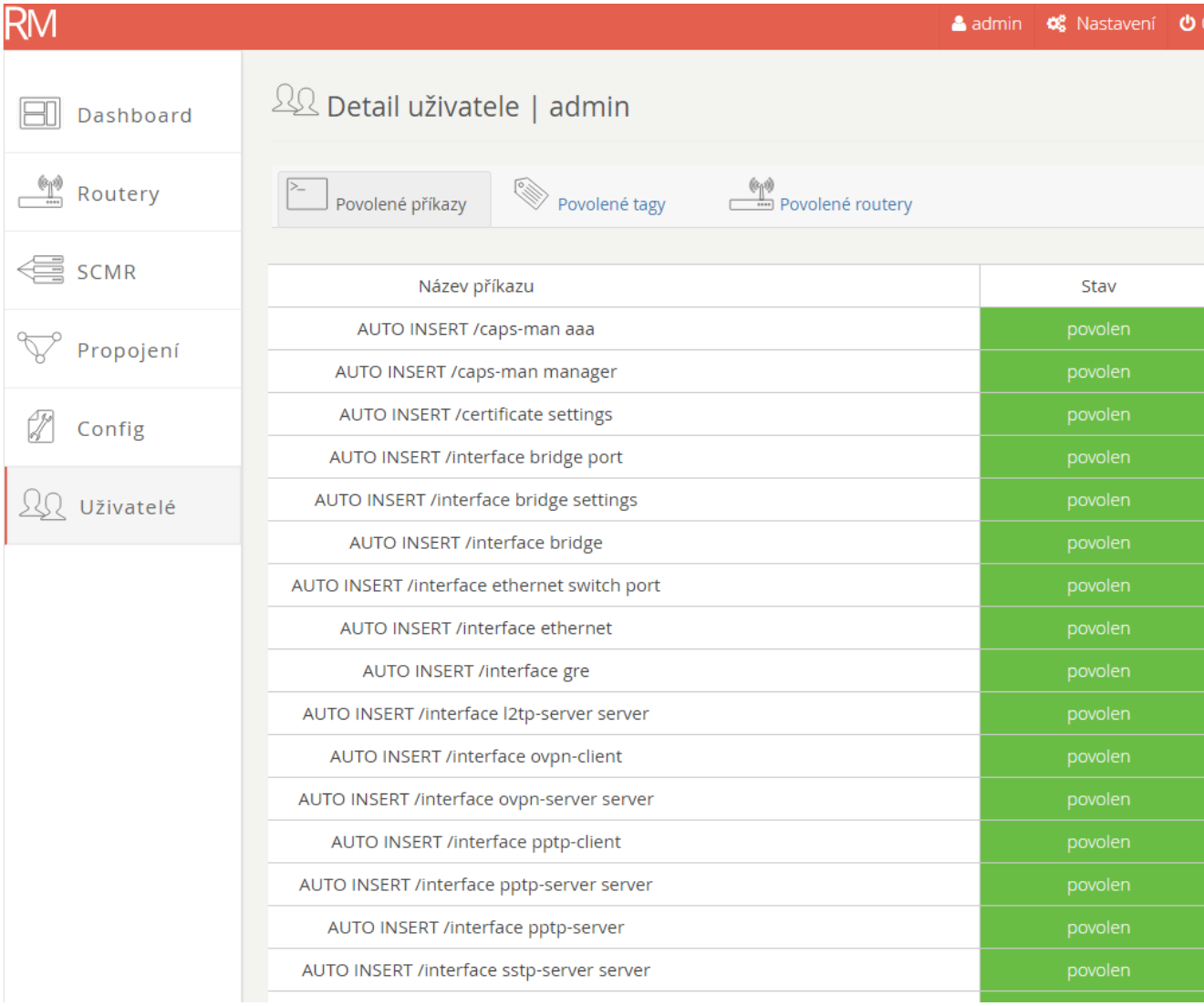
Velký důraz v této práci byl kladen také na zabezpečení celé aplikace. To spočívalo nejen v použití šifrovaného rozhraní MikroTik API-SSL, ale také bylo nutné implementovat do systému definice a správu přístupových oprávnění a přístupových skupin . Ve výchozím stavu má systém pouze jednoho funkčního uživatele. Tím je uživatel „admin@router.cz“ s heslem „123456“. Tento uživatel má v rámci systému neomezený přístup. Takový uživatel má stejnou funkcionalitu, jako uživatel root v operačním systému Linux. Další přídávání uživatelů je prováděno skrze záložku uživatelé v hlavním menu. Jakmile je uživatel přidán, je možné mu přiřadit jednotlivá oprávnění. Základem těchto práv je definice skupin zařízení, tzv. tagy a přiřazení jednotlivých povolených příkazů a zařízení.

Oprávnění v systému je děleno na příkazy, tagy a zařízení. Přednost má vždy právo odepření přístupu před právem přidělení přístupu. Nejdříve jsou aplikovány práva na skupinu zařízení. Pokud uživatel tedy přistupuje k zařízení, které je v nějaké skupině (tagu) a má povolen přístup, je pokračováno v dalším vyhodnocovaní. Pokud má přístup odepřen, má automaticky přístup odepřen i kdyby bylo samotné zařízení i samotný příkaz povolen. Pokud je však přístup povolen, je vyhodnoceno, zda má uživatel přístup k samotnému zařízení. Jestli že i toto vyhovuje, je zkontrolováno, zda má uživatel přístup k danému příkazu. Pokud ano, může provést nějakou editaci či přidaní nového příkazu.

Jestliže však je některe oprávnění zakázáno, uživatel má zařízení vždy v režimu read-only. Dokáže tedy konfiguraci vidět, ale nic neupraví. Jedno z možných nastavení si lze prohlédnout na následujícím obrázku.

2.3 Architektura MySQL databáze

Jak bylo řečeno, jako databázový systém byla vybrána databáze MySQL. Tato databáze je v současné době tvořena dvaceti čtyř tabulkami, které uchovávají všechny potřebné informace. Kódování databáze bylo zvoleno UTF8_unicode_ci a jako databázový úložný systém byl zvolen systém InnoDB, který je dnes standardním režimem MySQL. Kompletní databáze systému je vidět na následujícím obrázku.

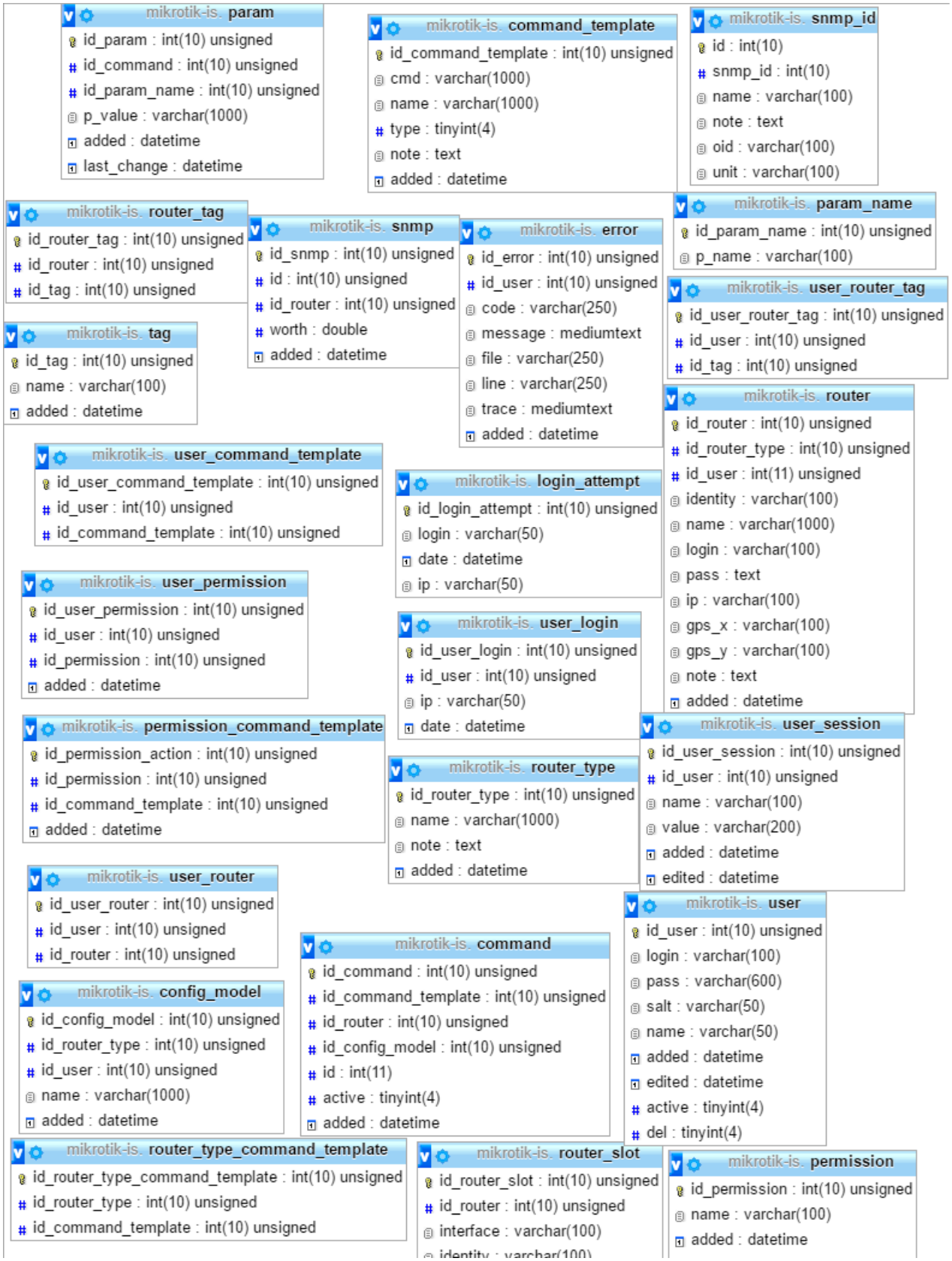


Obr. 2.4: Nastavení uživatelských práv k jednotlivým příkazům.

Hlavními tabulkami jsou tabulky `command` a `router`. Z nich jsou odvozeny všechny další. Zde uvádím popis pouze těch nejdůležitějších.

- Tabulka `command` - Tabulka s definicemi routerů
- Tabulka `command-template` - Tabulka s jednotlivými šablonami příkazů
- Tabulka `config-model` - Tabulka konfiguračních modelů
- Tabulka `param` - Tabulka s vloženými parametry příkazů na routeru
- Tabulka `router-tag` - Tabulka s přiřazením routerů do tagů
- Tabulka `snmp` - Tabulka se SNMP záznamy
- Tabulka `tag` - Tabulka definující jednotlivé tagy
- Tabulka `user` - Tabulka s uživateli
- Tabulka `user-router` - Tabulka s uživatelskými oprávněními k zařízením

- Tabulka `user-router-tag` - Tabulka s uživatelskými oprávněními ke skupinám zařízení

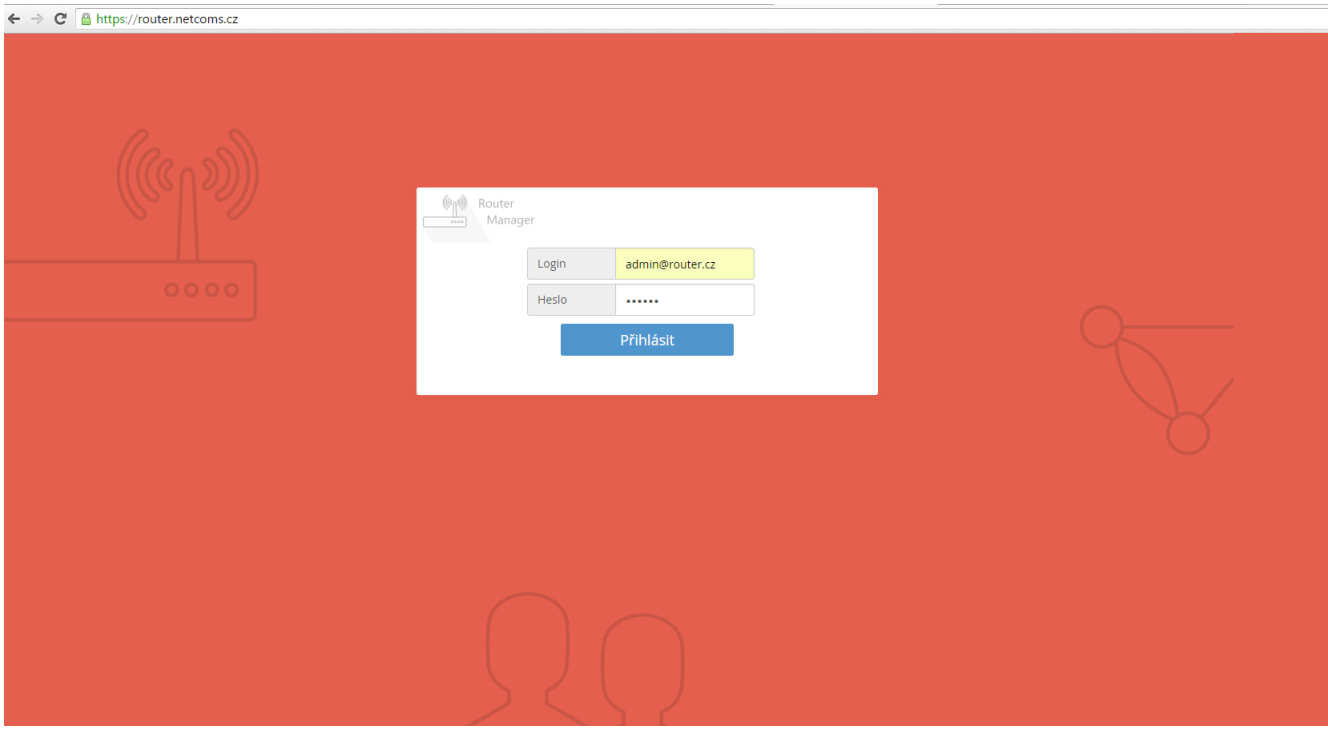


Obr. 2.5: Struktura MySQL databáze systému.

3 OVLÁDÁNÍ DALŠÍCH ČÁSTÍ SYSTÉMU

3.1 Přihlášení systému a doména běhu

Systém pro svůj běh vyžaduje běh na určité doméně. To se provede nastavením proměnné "WWW_ROOT v rámci souboru `./settings/frameworkSetting.php`. V rámci řešení práce byla použita doména router.netcoms.cz, na které probíhal jak vývoj, tak testování. Ve výchozím nastavení jsou přihlašovací údaje `admin@router.cz` a heslo `123456`. Na této doméně může být provedeno i finální testování funkčnosti vedoucím i oponentem práce. Doména bude zachována.



Obr. 3.1: Přihlášení do systému.

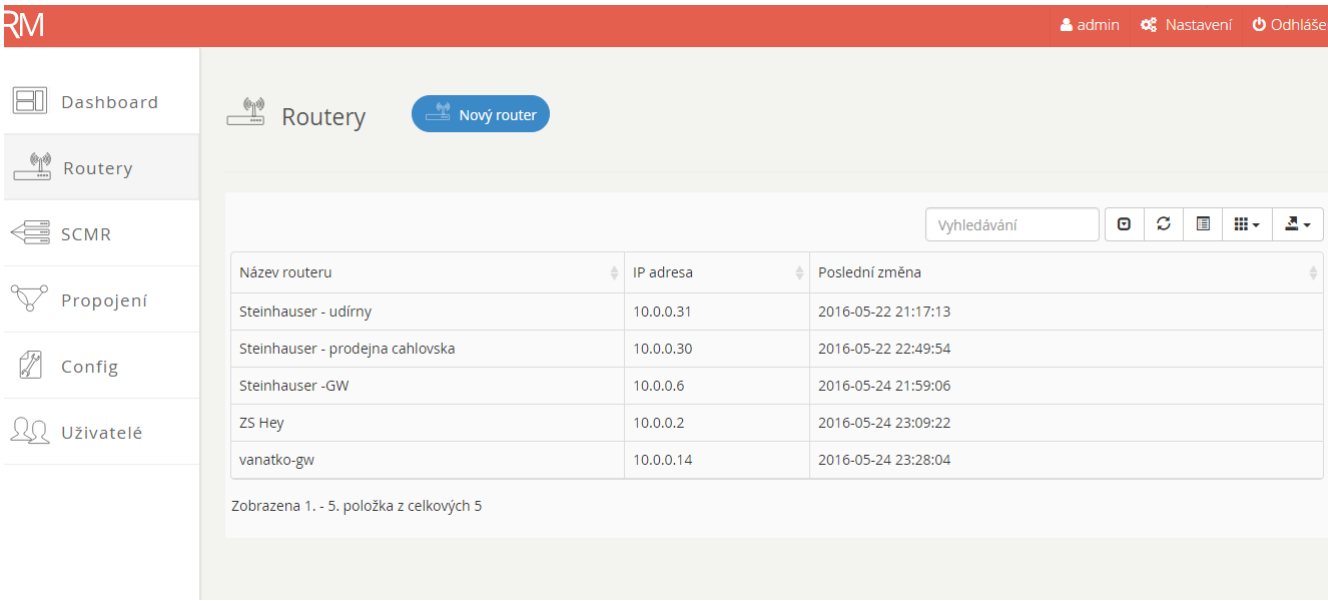
3.2 Přehled aplikace

Po přihlášení je uživateli nabídnuta obrazovka s hlavním přehledem, ve kterém najde všechna zařízení, která v systému existují. Je tedy aktivní karta `Routery` z hlavního menu. V této kartě může uživatel ihned router přidat.

Celé ovládání je koncipováno velmi jednoduše. V levém sloupci je hlavní menu pro přepínání na všechny části systému. Ve středu obrazovky se nachází ovládací plocha a vpravo nahoře tlačítko pro odhlášení, editaci uživatele a pro nastavení systému.

V rámci menu jsou k dispozici následující tlačítka:

- Dashboard - přehled zařízení a jejich dostupnost
- Routery - přehled routerů a možnost jejich editace po kliknutí na název případně přidání nového routeru
- SCMR - hromadné přidání jednoho či několika příkazů na více zařízení
- Propojení - graf propojení
- Config - správa konfiguračních vzorů
- Uživatelé - správa uživatelských rolí



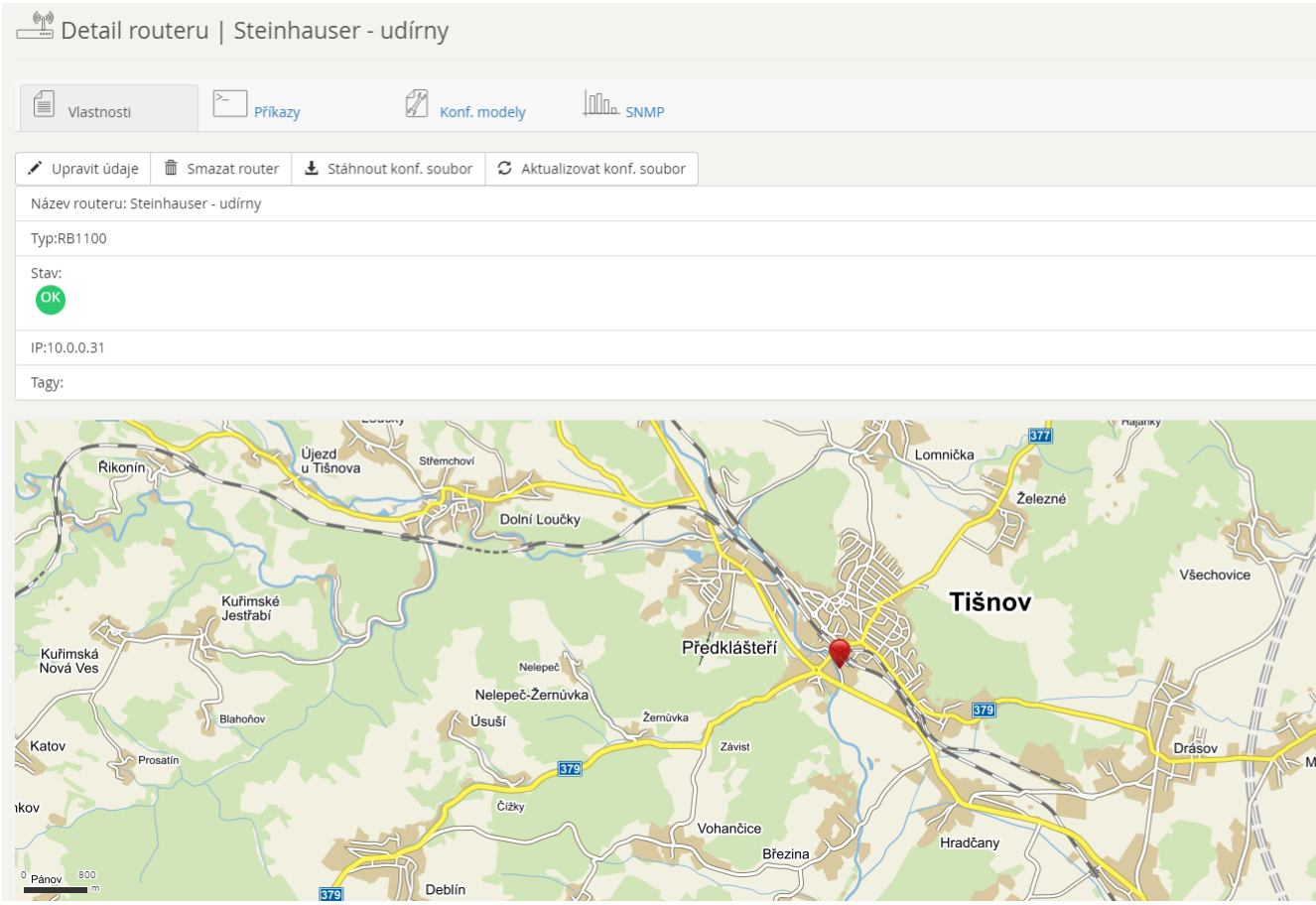
Obr. 3.2: Po přihlášení do systému.

3.3 Editace zařízení

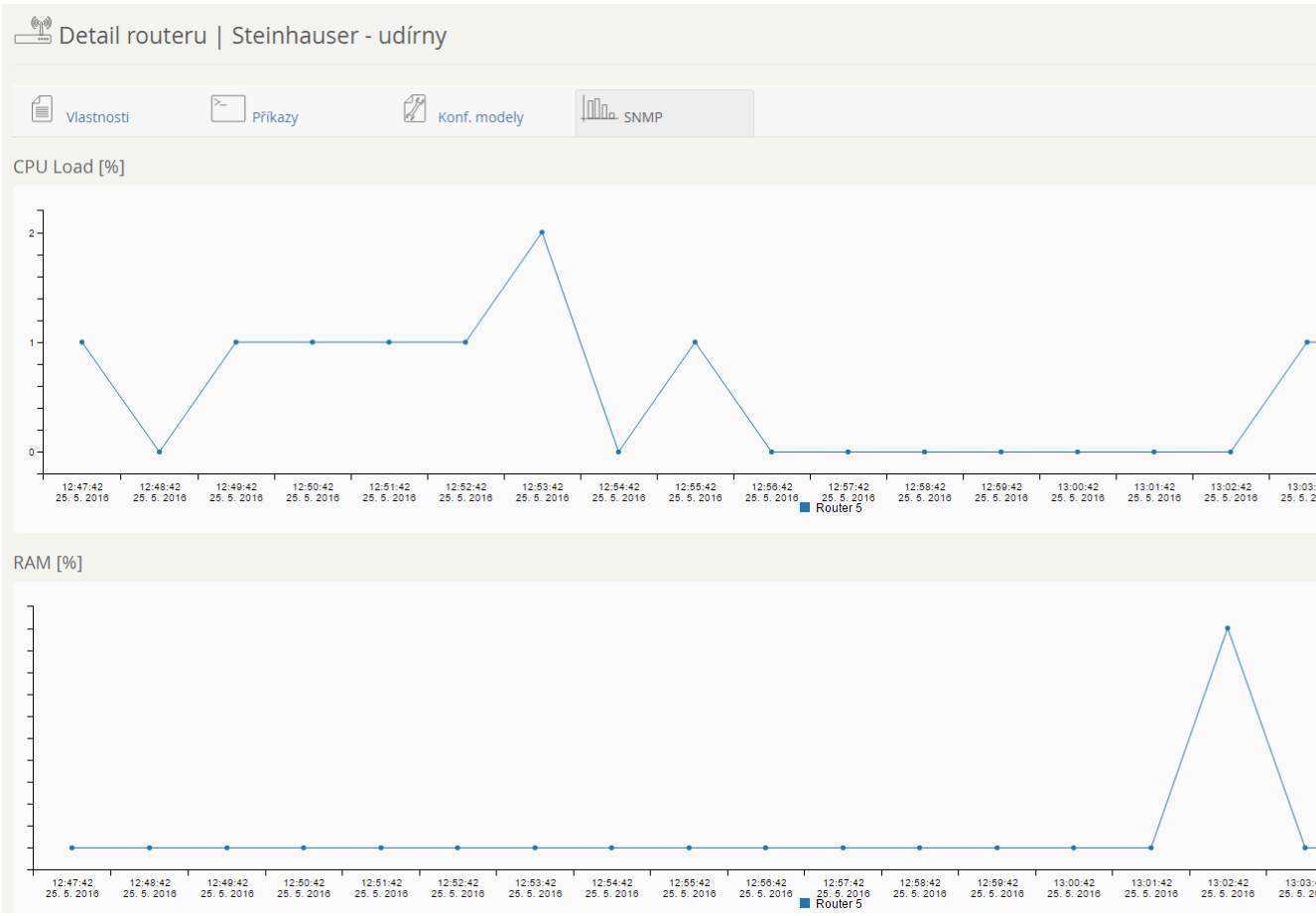
K editaci zařízení se lze dostat pomocí jednoduchého kliknutí na název daného zařízení. Po kliknutí na název dostaneme níže uvedený pohled. V něm můžeme provádět editaci zařízení, jeho konfiguraci nebo můžeme sledovat statistiky SNMP případně můžeme vidět zařízení na mapě, pokud byly zadány souřadnice GPS.

Jsou zde k dispozici následující volby:

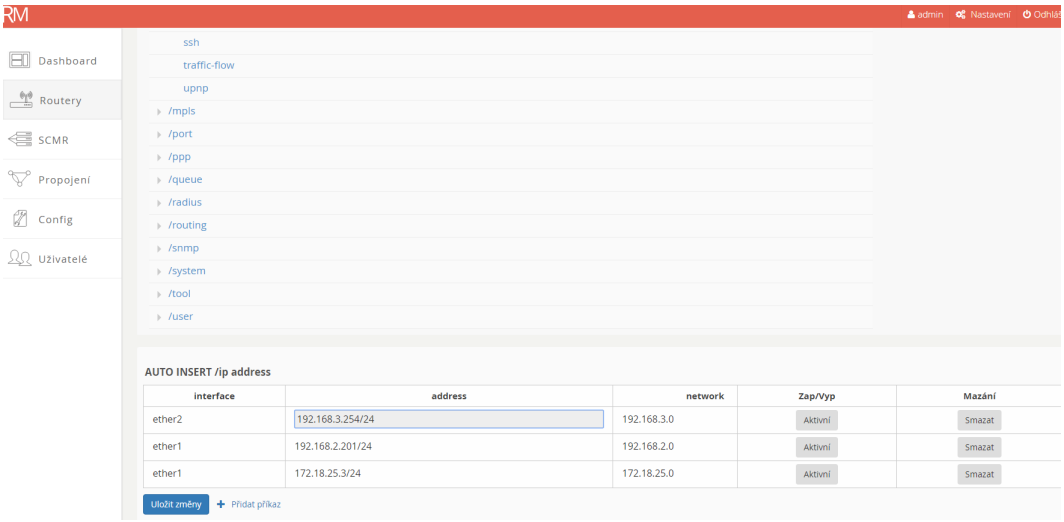
- Aktualizovat konf. soubor - import konfigurace z existujícího zařízení
- Stáhnout konf. soubor - stažení konfiguračního souboru jako textového souboru například na flash disk
- Smazat router - smazání zařízení ze systému
- SNMP - snmp statistiky a grafy
- Konf. modely - aplikace definovaných konfiguračnícho modelů
- Příkazy - online editace příkazů



Obr. 3.3: Editace zařízení



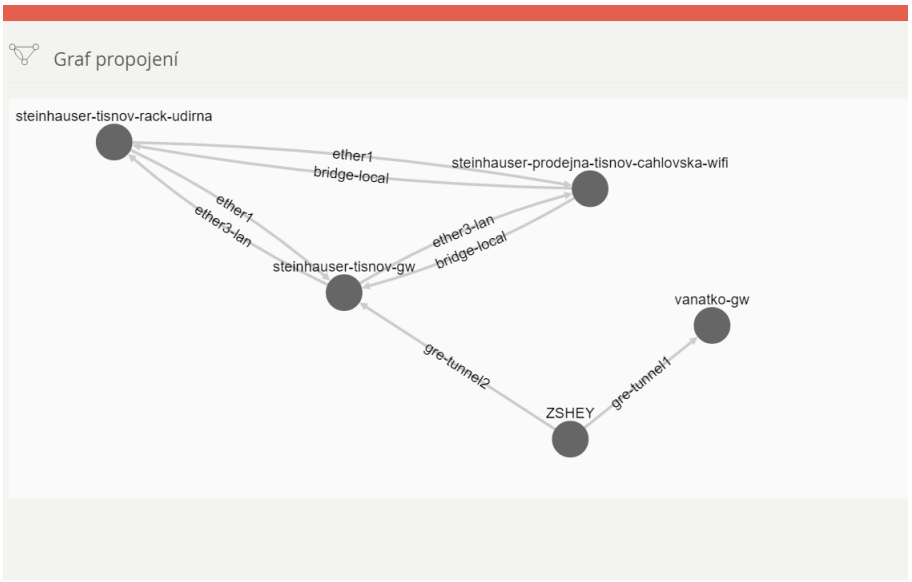
Obr. 3.4: SNMP statistiky



Obr. 3.5: Úprava konfigurace v reálném čase.

3.4 Graf propojení

Zobrazení aktuálního propojení a datových tras mezi prvky, které v systému jsou.



Obr. 3.6: Graf propojení

4 TESTOVÁNÍ FUNKČNOSTI

Testování každého produkčního nástroje je vždy velmi důležitým prvkem a aspektem celého vývoje softwarového projektu. Jelikož v této práci se jedná o aplikaci, která pro svoji funkčnost potřebuje reálná zařízení počítačové sítě, bylo trochu náročnější zřídit testovací prostředí pouze pro vývoj.

Ukázalo se, že testování v průběhu vývoje bylo velmi důležitým aspektem pro celou práci a odhalilo mnoho nečekaných chyb a problémů – například chybu interního číslování záznamů v RouterOS oproti číslování záznamů v externí MySQL tabulce při použití dynamických záznamů v konfiguraci (typicky IP adresa od DHCP klienta).

Samotné testování bylo nakonec rozvrženo do třech částí. První z nich se odehrávala při tvorbě semestrálního projektu v téměř ideálních podmínkách a při malé zátěži. Druhá v laboratorním prostředí větší sítě a třetí na reálných zařízení v reálně funkčních sítích po celé České republice. Testování bylo prováděno na prvcích různé cenové i výkonové kategorie a s různými místy určení a použití. Celé testovaní bylo přizpůsobeno tak, aby žádný z prvků nebyl ohrožen a žádné z testovaných zařízení nebylo ohroženo na svém chodu.

4.1 Použitá zařízení

Pro testování bylo použito celkem pět typů různých zařízení MikroTik Routerboard, každé umístěno v jiné lokalitě republiky. Všechny tyto zařízení jsou umístěna u mých klientů, kterým se starám o jejich počítačovou síť.

Zařízení jsou zapojena buď ve funkci centrálního směrovače pro celou jejich interní síť, z nějž jde následně linka do HW firewallu a dále k poskytovateli připojení, případně sami tvoří bránu mezi interní a externí sítí, včetně DHCP serveru a překladu adres NAT.

Všechny prvky měly instalován operační systém RouterOS verze 6.x. Mezi testovanými routerboardy nebyly žádné dva se stejnou verzí operačního systému.

Konkrétně šlo o tyto modely a umístění.

4.2 Popis testovací sítě

Jak je patrné z tabulky 4.1, většina zařízení se nacházejí mimo domov či zaměstnání autora práce. Je tedy zřejmé, že testování této aplikace muselo být rozděleno do dvou etap.

Tab. 4.1: Modely aktivních prvků použitých pro testování

Zařízení	Umístění	Verze OS
Routerboard RB750	Brno - byt autora práce	6.33.3
Routerboard RB450G	Trutnov - Obchodní akademie	6.26
Routerboard RB951Ui-2HnD	Brno - ZŠ Heyrovského 32	6.23
Routerboard RB1100AHx2	Brno - Gymnázium Bystrc	6.32.3
Routerboard RB941-2HnD	Skalice nad Svitavou - Steinhauser s.r.o.	6.25

Toto rozdělení bylo uděláno z toho důvodu, že aplikace zatím nepodporuje tzv. safe-mode, což je technologie RouterOS, která dokáže v případě jejího zapnutí detekovat ztrátu připojení uživatele k zařízení vlivem špatné změny konfigurace a vrátit nastavení zařízení do původního, tedy funkčního stavu. Pokud by tedy v rámci testování došlo k jakékoliv chybě, nejenže by chybu nebylo možno ihned odhalit tím, že by bylo zařízení přístupné z lokálního počítače komunikací po druhé vrstvě OSI modelu, ale hlavně by ji nebylo možné ihned opravit a zařízení vrátit do původní konfigurace. Toto by přineslo mnoho ztrát a to jak časových, neboť by bylo nutné k zařízení fyzicky dojít nebo dojet a přenastavit ho do funkčního stavu, ale také by tento problém přinesl mnoho ztrát finančních, neboť zakazníkova síť by byla nefunkční, což by znamenalo možné velké ztráty v jeho podnikání.

4.3 Průběh a problémy při tvorbě a testování

Z bezpečnostních a strategických důvodů a díky absenci funkce safe-mode tak bylo testování celé práce rozděleno do dvou etap, aby bylo možné co nejlépe předejít výše uvedeným možným problémům.

První etapa, kterou bylo možné spustit na všech zařízení, bylo ověření funkčnosti komunikace s API, její šifrování a funkčnosti autentizace a přihlášení k rozhraní. Z toho důvodu byl na všech zařízení zřízen účet uživatele „user1“, který měl přidělena práva pouze ke čtení. Bohužel, ve výsledku bylo zjištěno, že přístup skrze API rozhraní nefunguje správně, pokud má uživatel práva pouze ke čtení, takže bylo nutné změnit práva na plný přístup. Zároveň bylo nutné na zařízení vygenerovat SSL certifikát, který mohl být klidně podepsaný sám sebou a přiřadit ho službě API-SSL, aby mohlo vůbec k šifrování dojít. Systém má nastaveno to, že nemá kontrolovat validitu certifikátů, takže je možné použít téměř jakýkoliv certifikát. V tomto případě bylo zvoleno vygenerování podepsaného certifikátu sebou samým s platností 50 let.

Tímto byla ověřena funkčnost základní komunikace, vyčítání základních hodnot ze zařízení a otestování celého přenosu skrze rozhraní API, které bylo šifrované

pomocí SSL. V této fázi tak byl ověřen přístup na zařízení a možnost provádět se zařízením všechny věci, které jsou potřeba.

Druhá etapa testování už byla prováděna pouze na zařízení Routerboard RB750, které bylo umístěno v bytě autora práce a v případě potřeby bylo jednoduché jeho konfiguraci opět velmi rychle obnovit do původního stavu prostřednictvím nástroje WinBox a komunikací na druhé vrstvě ISO/OSI modelu.

Zde byl opět uživatel „user1“ povolen i zápis. Testování přenosu konfiguračních souborů a jejich zápis tak bylo uskutečňováno pouze na tomto zařízení. V průběhu testování nebyl odhalena jediná nekompatibilita či rozdílnost výstupů oproti očekávání.

Závěrečné testování proběhlo také na směrovači Základní školy Brno, Heyrovského 32. Vzhledem k rychlé dostupnosti v případě nesprávné konfigurace nebyl velký problém ke směrovači dojít a komunikaci obnovit. Naštěstí, závěrečný test dopadl bez problémů a ukázal funkční nejen konfiguraci na základě konfiguračních vzorů, ale také konfiguraci v reálném čase.

4.4 Zhodnocení

Po provedeném testování je možné říci, že systém nevykazuje kromě malých detailů žádnou větší vadu, která by se týkala komunikace se zařízeními MikroTik.

5 ZÁVĚR

V rámci této moji semestrální práce bylo úspěšně vytvořeno prostředí a základ webového informačního systému pro správu počítačových sítí, které jsou postaveny nad prvky společnosti MikroTik . Tato práce tedy dává základ pro velmi robustní systém pro správu zařízení platformy MikroTik a nabízí možnosti budoucího rozvoje do komerčního či opensource projektu.

5.1 Aktuální stav

Nyní je aplikace plně použitelná, avšak, funkčnost a možnosti jsou na cca 90% optimální funkčnosti potřebné pro nasazení v reálném provozu. V práci chybí některé ochranné věci a věci týkající se intuice ovládaní.

V současné době aplikace podporuje kompletně přidání a odebrání nových zařízení do systému, správu uživatelů, správu uživatelských rolí, tvorbu a změny konfiguračních šablon.

Zařízení je tedy možno konfigurovat třemi způsoby – hromadnou změnou příkazů se společnými parametry (typicky hesla, fronty, VLAN sítě apod.), nahrazením aktuální konfigurace celým konfiguračním obrazem nebo úpravou jednotlivých konfiguračních parametrů v reálném čase. Všechny tyto možnosti tedy poskytují základní funkcionalitu, která je dostatečná pro jednoduchou správu rozlehlých sítí.

Celý systém je postaven velmi dynamicky. Pro přidání podpory těchto balíčků stačí opět pouze správně v databázi nadefinovat podobu příkazů a systém na základě nich vytvoří správné rozložení GUI a umožní uživateli editaci a správu těchto nových voleb, stejně jako jejich vyčtení ze zařízení a zápis nových konfigurací do něj.

5.2 Nedostatky řešení

Aktuální řešení má však také několik nedostatků. Jsou jimi zejména neošetřené některé možné situace, které v RouterOS mohou nastat.

Toto chování bude opraveno v příští verzi.

5.3 Budoucí rozšíření práce

Práce bude dále rozšiřována do podoby firemního informačního systému. Budou opraveny nejen všechny chyby, které existují či budou nalezeny, ale systém bude rozšířen o mnoho dalších funkcionalit.

Jednou z nejvýznamnějších funkcí, které budou v rámci řešení do systému doplněny, je jednoznačně funkce tzv. safe-mode. Jak již bylo řečeno v kapitole věnované testování, safe-mode je funkcionalita, která rozpozná odpojeného administrátora zařízení z důvodu špatně provedené konfigurační změny. Bez této funkce by bylo nutné dojít k vybranému aktivnímu prvku fyzicky, restartovat ho do továrního nastavení, bylo by nutné ho znovu zavést do systému a znovu nakonfigurovat. V případě, že bude zaplý safe-mode, RouterOS automaticky rozpozná, že došlo k odpojení administrátora vlivem špatné konfigurace a vrátí všechny konfigurační kroky, které administrátor provedl během přihlášené relace, obnoví tak předešlé funkční nastavení. Není tedy nutné k zařízení chodit, ani mu mazat konfiguraci.

Systém bude také dále obsahovat modul pro správu koncových zákazníků poskytovatele, kteří od něj budou mít koncový bod se systémem RouterOS. Jako poslední velké vylepšení bude naprogramována plná podpora balíku IPv6 pro možnost konfigurace zařízení nejen prostřednictvím IPv6, ale také pro přípravu celé sítě na IPv6.

LITERATURA

[1] Kolektiv autorů *Kompletní manuálové stránky společnosti Mikrotik* [online]. 2015, poslední aktualizace 30. 06. 2015 [cit. 25. 05. 2016]. Dostupné z URL: <`http://wiki.mikrotik.com/wiki/Manual:TOC`>.

[2] Kolektiv autorů *Kompletní manuálové stránky jazyka PHP* [online]. 2015, poslední aktualizace 13. 12. 2015 [cit. 25. 05. 2016]. Dostupné z URL: <`https://secure.php.net/manual/en/`>.

[3] Kolektiv autorů *Popis funkčnosti API rozhraní zařízení Mikrotik* [online]. 2015, poslední aktualizace 02. 09. 2015 [cit. 25. 05. 2016]. Dostupné z URL: <`http://wiki.mikrotik.com/wiki/Manual:API`>.

[4] Kolektiv autorů *Mikrotik API klient v PHP* [online]. 2015, poslední aktualizace 19. 11. 2015 [cit. 25. 05. 2016]. Dostupné z URL: <`https://github.com/BenMenking/routeros-api`>.

[5] Kolektiv autorů *Produktový katalog Routerboard* [online]. 2015, poslední aktualizace 19. 11. 2015 [cit. 25. 05. 2016]. Dostupné z URL: <`http://routerboard.com/`>.

[6] Kolektiv autorů *Mikrotik API klient v PHP* [online]. 2015, poslední aktualizace 31. 08. 2015 [cit. 25. 05. 2016]. Dostupné z URL: <`http://wiki.mikrotik.com/wiki/API_PHP_class`>.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

PHP	PHP: Hypertext Preprocessor - PHP: Hypertextovy preprocesor
ISP	Internet Service Provider - poskytovatel internetového připojení
MySQL	Varianta jazyka SQL - Structured Query Language
API	Application Programming Interface - rozhraní pro programování aplikací
SSL	Secure Sockets Layer - vrstva bezpečných socketů - rozrahní pro zabezpečenou síťovou komunikaci
GUI	Graphical User Interface - grafické uživatelské rozhraní
LAMP	Linux Apache MySQL PHP - balík aplikací pro běh webových stránek
DHCP	Dynamic Host Configuration Protocol - protokol pro automatickou konfiguraci síťových parametrů v prostředí IPv4
NAT	Network Address Translation - mechanismus pro překlad síťovych IPv4 adres
HW	Hardware - fyzické části počítače či prvku
SFTP	SSH File Transfer Protocol - protokol a program pro bezpečný přenos souborů pomocí počítačové sítě prostřednictvím SSH
SSH	Secure shell - zabezpečený protokol a program pro připojení ke vzdálenému terminálu či pro přenos souborů prostřednictvím počítačové sítě
TCP	Transmission Control Protocol - protokol transportní vrstvy ISO/OSI modelu zaručující spolehlivé spojení mezi dvěma komunikujícími zařízeními